Assignment 2

Q1. In this case for a smaller training data set I expect LDA to work better since the covariance matrice cov__1 and cov__2 are equal and LDA with p>1 assumes equal covariances. Furthermore, there is a limited training data set so reducing variance is crucial which LDA accomplishes. In comparison QDA might not work as well on a smaller training set as QDA is more flexible than LDA but tends to overfit on smaller training data. In the case for a larger training data set I expect QDA to work better because of its flexibility with higher dimensions and datasets.

Q2. As observed the LDA is better (has higher accuracy) than QDA for smaller datasets. In comparison QDA works better for larger datasets which is in line with my expectations.

#Install packages

```r
library(MASS)
library(rmarkdown)
library(tinytex)
```

#Generate Data Function

```r
gen_data <- function(n) {
  p <- 3
  n1 <- n2 <- n/2
  cov_1 <- diag(rep(1,p)) + 0.2
  cov_2 <- cov_1
  cov_2[1,2] <- cov_2[2,1] <- cov_2[1,2] + 0.5
  x_class1 <- mvrnorm(n1, mu = rep(3,p), Sigma = cov_1)
  x_class2 <- mvrnorm(n2, mu = rep(2,p), Sigma = cov_2)
  x <- rbind(x_class1, x_class2)
  y <- rep(c(1,2), c(n1, n2))
  df <- as.data.frame(cbind(x,y))
  names(df) <- c(paste0("x", 1:p), "y")
  return(df)
}
```

#LDA for training set of size 50

```r
sum_lda_mean_50 <- 0


for (i in 1:100){
  set.seed(123)
  train_set_50 <- gen_data(50)
  test_set_10000 <- gen_data(10000)

  lda.fit <- lda(y ~ x1 + x2 + x3, data = train_set_50)

  lda.pred <- predict(lda.fit, test_set_10000)

  lda_mean_50 <- mean(lda.pred$class == test_set_10000$y)
  sum_lda_mean_50 <- sum_lda_mean_50 + lda_mean_50
}

avg_lda_mean_50 <- sum_lda_mean_50/100

print(avg_lda_mean_50)
```

```
## [1] 0.7226
```

#LDA with training set of size 10000

```r
sum_lda_mean_10000 <- 0
for (i in 1:100){
  set.seed(123)
  #TRAINING SET OF SIZE 10000
  train_set_10000 <- gen_data(10000)
  test_set_10000 <- gen_data(10000)

  lda.fit <- lda(y ~ x1 + x2 + x3, data = train_set_10000)
  lda.pred <- predict(lda.fit, test_set_10000)
  lda_mean_10000 <- mean(lda.pred$class == test_set_10000$y)

  #print(mean_10000)
  sum_lda_mean_10000 <- sum_lda_mean_10000 + lda_mean_10000
}


avg_lda_mean_10000 <- sum_lda_mean_10000/100
print(avg_lda_mean_10000)
```

```
## [1] 0.7384
```

```r
#avg_mean_50 = sum(sum_mean_50)/100
#print(avg_mean_10000)
```

#QDA with training set of size 50

```r
sum_qda_mean_50 <- 0


for (i in 1:100){
  set.seed(123)
  train_set_50 <- gen_data(50)
  test_set_10000 <- gen_data(10000)

  qda.fit <- qda(y ~ x1 + x2 + x3, data = train_set_50)

  qda.pred <- predict(qda.fit, test_set_10000)

  mean_50 <- mean(qda.pred$class == test_set_10000$y)
  sum_qda_mean_50 <- sum_qda_mean_50 + mean_50

}

avg_qda_mean_50 <- sum_qda_mean_50/100

print(avg_qda_mean_50)
```

```
## [1] 0.7204
```

#QDA with a training set of size 10000

```
sum_qda_mean_10000 <- 0
for (i in 1:100){
  set.seed(123)
  train_set_10000 <- gen_data(10000)
  test_set_10000 <- gen_data(10000)

  qda.fit <- qda(y ~ x1 + x2 + x3, data = train_set_10000)
  qda.pred <- predict(qda.fit, test_set_10000)
  mean_10000 <- mean(qda.pred$class == test_set_10000$y)

  sum_qda_mean_10000 <- sum_qda_mean_10000 + mean_10000
}


avg_qda_mean_10000 <- sum_qda_mean_10000/100
```

```
##         50  10000
## LDA 0.7226 0.7384
## QDA 0.7204 0.7495
```